2-State 2-Symbol Turing Machines with Periodic Support Produce Regular Sets

Neary, Turlough

Abstract: We say that a Turing machine has periodic support if there is an infinitely repeated word to the left of the input and another infinitely repeated word to the right. In the search for the smallest universal Turing machines, machines that use periodic support have been significantly smaller than those for the standard model (i.e. machines with the usual blank tape on either side of the input). While generalising the model allows us to construct smaller universal machines it makes proving decidability results for the various state-symbol products that restrict program size more difficult. Here we show that given an arbitrary 2-state 2-symbol Turing machine and a configuration with periodic support the set of reachable configurations is regular. Unlike previous decidability results for 2-state 2-symbol machines, here we include in our consideration machines that do not reserve a transition rule for halting, which further adds to the difficulty of giving decidability results.

DOI: https://doi.org/10.1007/978-3-319-60252-3_22

Originally published at:
Neary, Turlough (2017). 2-State 2-Symbol Turing Machines with Periodic Support Produce Regular Sets. In: 19th International Conference on Descriptional Complexity of Formal Systems, Milan, 3 July 2017 - 5 July 2017. Springer, 274-286.
DOI: https://doi.org/10.1007/978-3-319-60252-3_22

2-state 2-symbol Turing Machines with Periodic Support Produce Regular Sets

Turlough Neary

Institute of Neuroinformatics
University of Zurich and ETH Zurich

DCFS, 2017

Introduction

- Shannon (58') asks what is the smallest state-symbol product needed to give a single-tape universal Turing machine (UTM).
- Over the years many authors took up Shannon's challenge to find the smallest UTM. (Minsky, Rogozhin, Kudlek, Neary and Woods)
- By allowing periodic support we can find much smaller universal machines. (Watanabe, Cook, Neary and Woods)
  - Existing decidability results not applicable.
  - Makes the problem of proving decidability results more difficult.

Main result

A TM with periodic support has a blank word repeated to left of its input and another repeated to the right.

Theorem

Given an arbitrary 2-state, 2-symbol, single-tape Turing machine with periodic support, the set of reachable configurations from an arbitrary configuration is regular.

Many questions for such TMs are now decidable

- Will a computation halt?
- Does word w appear on the tape during a computation?
- Does a computation enter a repeating sequence of configurations?

Turing machines with periodic support

Definition

A Turing machine with periodic support is a quintuple $M = (Q, \Sigma, f, l, r)$:

- state set $Q = \{q_1, \ldots q_{|Q|}\}$
- symbol set $\Sigma = \{\sigma_1, \ldots \sigma_{|\sigma|}\}$
- transition function $f : Q \times \Sigma \to \Sigma \times \{L, R\} \times Q$
- left and right blank words $l, r \in \Sigma^+$

No start state and no halting rule.

Turing machines with periodic support

TM configuration: $c = u\,\mathbf{q_x}\sigma\,v$

- q_x is the current state
- σ is the read symbol
- $u, v \in \Sigma^*$ are the words to the left and right of σ on the tape

Appended blank word $r \in \{0,1\}^*$ to c if the head exits to the right.
Prepended blank word $l \in \{0,1\}^*$ to c if the head exits to the left.

Computation step: $c_1 \vdash c_2$
0 or more computations steps: $c_1 \vdash^* c_t$
Configurations reachable from c_1: $\{c_i | c_1 \vdash^* c_i\}$

2-state 2-symbol Turing machines

An arbitrary 2-state 2-symbol TM is given by the 4 transition rules:

$$q_a, 0, \sigma_1, d_1, q_1$$
$$q_a, 1, \sigma_2, d_2, q_2$$
$$q_b, 0, \sigma_3, d_3, q_3$$
$$q_b, 1, \sigma_4, d_4, q_4$$

Here $\sigma_i \in \{0, 1\}$ is the write symbol, $d_i \in \{R, L\}$ is the move direction and $q_i \in \{q_a, q_b\}$ is the next state.

We can denote each possible machine as a triple (Σ, D, Q) where $\Sigma = \{\sigma_1, \sigma_2, \sigma_3, \sigma_4\}$, $D = \{d_1, d_2, d_3, d_4\}$, and $Q = \{q_1, q_2, q_3, q_4\}$.

Reducing 4096 cases to 328 cases

$$\begin{array}{rcccccc} & \Sigma_1 & \Sigma_2 & \Sigma_3 & \Sigma_4 & \Sigma_5 & \Sigma_6 \\ \sigma_1 = & 0 & 0 & 0 & 0 & 0 & 1 \\ \sigma_2 = & 0 & 0 & 0 & 0 & 1 & 0 \\ \sigma_3 = & 0 & 0 & 1 & 1 & 1 & 1 \\ \sigma_4 = & 0 & 1 & 0 & 1 & 0 & 0 \end{array}$$

The 16 possible cases for the 4 write symbols reduced to 6 cases.

$$\begin{array}{rccccccc} & D_1 & D_2 & D_3 & D_4 & D_5 & D_6 & D_7 \\ d_1 = & L & L & L & R & L & L & L \\ d_2 = & L & L & R & L & L & R & R \\ d_3 = & L & R & L & L & R & L & R \\ d_4 = & R & L & L & L & R & R & L \end{array}$$

The 16 possible cases for the 4 move values reduced to 7 cases.

$$\begin{array}{rccccccccc} & Q_1 & Q_2 & Q_3 & Q_4 & Q_5 & Q_6 & Q_7 & Q_8 & Q_9 \\ q_1 = & q_a & q_b & q_a & q_a & q_b & q_b & q_b & q_b & q_b \\ q_2 = & q_b & q_a & q_b & q_b & q_a & q_a & q_b & q_b & q_b \\ q_3 = & q_a & q_a & q_a & q_b & q_a & q_b & q_a & q_a & q_b \\ q_4 = & q_a & q_a & q_b & q_a & q_b & q_a & q_a & q_b & q_a \end{array}$$

The 16 possible cases for the 4 next state values reduced to 9 cases.

Regularity of $\{c_i | c_1 \vdash c_i\}$ is closed under 0-1, R-L and q_a-q_b symmetries.

Reducing 4096 cases to 328 cases

	Σ_1	Σ_2	Σ_3	Σ_4	Σ_5	Σ_6
$\sigma_1 =$	0	0	0	0	0	1
$\sigma_2 =$	0	0	0	0	1	0
$\sigma_3 =$	0	0	1	1	1	1
$\sigma_4 =$	0	1	0	1	0	0

The 16 possible cases for the 4 write symbols reduced to 6 cases.

	D_1	D_2	D_3	D_4	D_5	D_6	D_7
$d_1 =$	L	L	L	R	L	L	L
$d_2 =$	L	L	R	L	L	R	R
$d_3 =$	L	R	L	L	R	L	R
$d_4 =$	R	L	L	L	R	R	L

The 16 possible cases for the 4 move values reduced to 7 cases.

	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	Q_9
$q_1 =$	q_a	q_b	q_a	q_a	q_b	q_b	q_b	q_b	q_b
$q_2 =$	q_b	q_a	q_b	q_b	q_a	q_a	q_b	q_b	q_b
$q_3 =$	q_a	q_a	q_a	q_b	q_a	q_b	q_a	q_a	q_b
$q_4 =$	q_a	q_a	q_b	q_a	q_b	q_a	q_a	q_b	q_a

The 16 possible cases for the 4 next state values reduced to 9 cases.

Regularity of $\{c_i | c_1 \vdash c_i\}$ is closed under 0-1, R-L and q_a-q_b symmetries.
Remove symmetries.

Reducing 4096 cases to 328 cases

	Σ_1	Σ_2	Σ_3	Σ_4	Σ_5	Σ_6
$\sigma_1 =$	0	0	0	0	0	1
$\sigma_2 =$	0	0	0	0	1	0
$\sigma_3 =$	0	0	1	1	1	1
$\sigma_4 =$	0	1	0	1	0	0

The 16 possible cases for the 4 write symbols reduced to 6 cases.

	D_1	D_2	D_3	D_4	D_5	D_6	D_7
$d_1 =$	L	L	L	R	L	L	L
$d_2 =$	L	L	R	L	L	R	R
$d_3 =$	L	R	L	L	R	L	R
$d_4 =$	R	L	L	L	R	R	L

The 16 possible cases for the 4 move values reduced to 7 cases.

	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	Q_9
$q_1 =$	q_a	q_b	q_a	q_a	q_b	q_b	q_b	q_b	q_b
$q_2 =$	q_b	q_a	q_b	q_b	q_a	q_a	q_b	q_b	q_b
$q_3 =$	q_a	q_a	q_a	q_b	q_a	q_b	q_a	q_a	q_b
$q_4 =$	q_a	q_a	q_b	q_a	q_b	q_a	q_a	q_b	q_a

The 16 possible cases for the 4 next state values reduced to 9 cases.

Regularity of $\{c_i | c_1 \vdash c_i\}$ is closed under 0-1, R-L and q_a-q_b symmetries.

Remove symmetries. Remove trap states.

Reducing 4096 cases to 328 cases

Q_7	Q_8	Q_9
Q_4	Q_5	Q_6
Q_1	Q_2	Q_3

$\longleftarrow$ Location of each Q_i case within each (Σ_i, D_j) 3×3 block below

Σ_6	B	C	B	A	A	A	A	A	A	A	A	A	B	A	A	C	A	A	C	A	A
	B	C	B	A	A	A	A	A	A	A	A	A	C	A	A	C	A	A	B	A	A
	B	B	B	A	A	A	A	A	A	A	A	A	B	B	C	C	C	C	B	B	B
Σ_5	B	B	B	A	A	A	B	B	B	A	A	A	B	B	A	B	B	A	B	B	A
	B	B	B	A	A	A	B	B	B	A	A	A	B	A	A	B	A	A	B	A	A
	B	B	B	A	A	A	B	B	B	A	A	A	B	A	B	B	A	E	B	A	C
Σ_4	B	B	B	B	B	B	A	A	A	A	A	A	B	A	A	B	A	A	B	A	A
	B	B	B	B	B	B	A	A	A	A	A	A	C	A	C	B	A	C	E	A	B
	B	B	B	B	B	B	A	A	A	A	A	A	B	B	B	B	B	B	B	B	B
Σ_3	B	B	B	B	B	C	B	B	B	B	B	B	B	B	B	B	B	C	B	B	B
	B	B	B	C	B	C	B	B	B	B	B	B	C	B	B	C	B	C	B	B	B
	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
Σ_2	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
	B	B	B	C	B	B	B	B	B	B	B	B	B	B	B	B	B	B	C	B	B
	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
Σ_1	B	B	B	B	B	B	A	A	A	A	A	A	B	B	B	B	B	B	B	B	B
	B	B	B	C	B	B	A	A	A	A	A	A	C	B	B	B	B	B	C	B	B
	B	B	B	B	B	B	A	A	A	A	A	A	B	B	B	B	B	B	B	B	B
		D_1			D_2			D_3			D_4			D_5			D_6			D_7	

Case A:
reduction to
symmetric case

Case B:
periodic computation

Case C:
semi-periodic computation

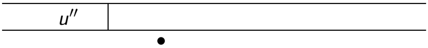
Case E:
binary counter

Periodic computations produce regular sets

Transition rule sequence s is executed between times $t + i|s|$ and $t + (i + 1)|s|$ for all $i \in \mathbb{N}$.

We need only consider that case where s contains m more right move than left move instructions for some $m \in \mathbb{N}$.

Computation beginning at time t



$\longleftarrow$ TM tape

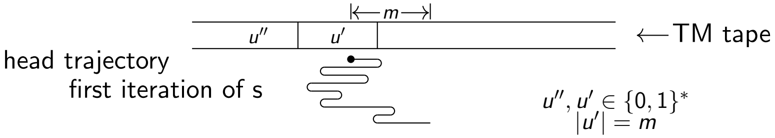
$u'', u' \in \{0, 1\}^*$

Periodic computations produce regular sets

Transition rule sequence s is executed between times $t + i|s|$ and $t + (i+1)|s|$ for all $i \in \mathbb{N}$.

We need only consider that case where s contains m more right move than left move instructions for some $m \in \mathbb{N}$.

Computation beginning at time t

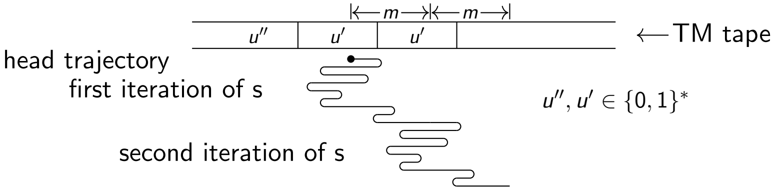


Periodic computations produce regular sets

Transition rule sequence s is executed between times $t + i|s|$ and $t + (i+1)|s|$ for all $i \in \mathbb{N}$.

We need only consider that case where s contains m more right move than left move instructions for some $m \in \mathbb{N}$.

Computation beginning at time t

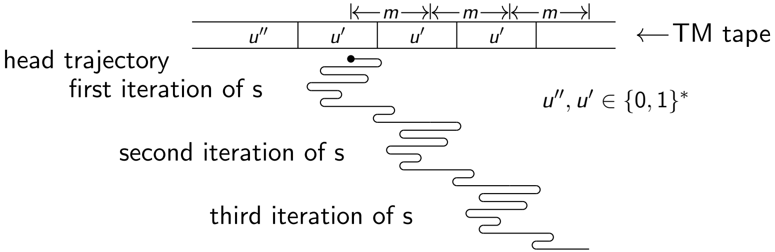


Periodic computations produce regular sets

Transition rule sequence s is executed between times $t + i|s|$ and $t + (i + 1)|s|$ for all $i \in \mathbb{N}$.

We need only consider that case where s contains m more right move than left move instructions for some $m \in \mathbb{N}$.

Computation beginning at time t

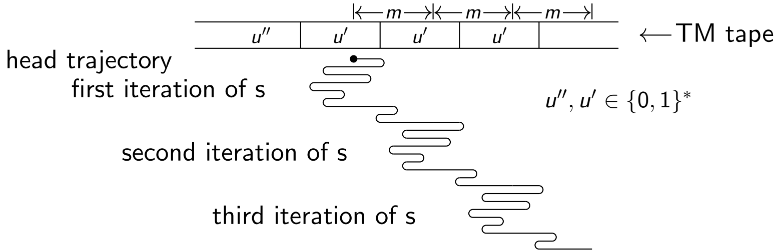


Periodic computations produce regular sets

Transition rule sequence s is executed between times $t + i|s|$ and $t + (i+1)|s|$ for all $i \in \mathbb{N}$.

We need only consider that case where s contains m more right move than left move instructions for some $m \in \mathbb{N}$.

Computation beginning at time t



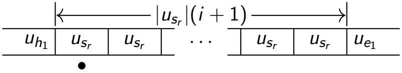
Pumping the number of u' words gives all reachable configurations.

Semi-periodic computations produce regular sets

After time $t(0)$ the sequence S_i is executed between times $t(i)$ and $t(i+1)$ for all $i \in \mathbb{N}$, where $t(i+1) = t(i) + |S_i|$.

$$S_i = (s_r)^{i+1} e_1 (s_l)^{i+1} h_1 (s_r)^{i+1} e_2 (s_l)^{i+1} h_2 \ldots (s_r)^{i+1} e_x (s_l)^{i+1} h_x$$

$u_{s_r}, u_{s_l}, u_{h_j}, u_{e_j} \in \{0,1\}^*$

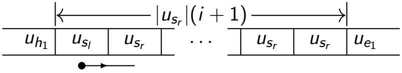


Semi-periodic computations produce regular sets

After time $t(0)$ the sequence S_i is executed between times $t(i)$ and $t(i+1)$ for all $i \in \mathbb{N}$, where $t(i+1) = t(i) + |S_i|$.

$$S_i = (s_r)^{i+1} e_1 (s_l)^{i+1} h_1 (s_r)^{i+1} e_2 (s_l)^{i+1} h_2 \ldots (s_r)^{i+1} e_x (s_l)^{i+1} h_x$$

$u_{s_r}, u_{s_l}, u_{h_j}, u_{e_j} \in \{0,1\}^*$



s_r contains m more right moves than left moves

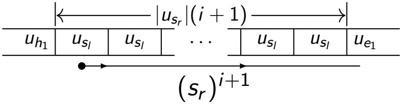
$|u_{s_r}| = |u_{s_l}| = m$

Semi-periodic computations produce regular sets

After time $t(0)$ the sequence S_i is executed between times $t(i)$ and $t(i+1)$ for all $i \in \mathbb{N}$, where $t(i+1) = t(i) + |S_i|$.

$$S_i = (s_r)^{i+1} e_1 (s_l)^{i+1} h_1 (s_r)^{i+1} e_2 (s_l)^{i+1} h_2 \ldots (s_r)^{i+1} e_x (s_l)^{i+1} h_x$$

$u_{s_r}, u_{s_l}, u_{h_j}, u_{e_j} \in \{0,1\}^*$



s_r contains m more right moves than left moves

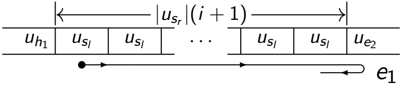
$|u_{s_r}| = |u_{s_l}| = m$

Semi-periodic computations produce regular sets

After time $t(0)$ the sequence S_i is executed between times $t(i)$ and $t(i+1)$ for all $i \in \mathbb{N}$, where $t(i+1) = t(i) + |S_i|$.

$$S_i = (s_r)^{i+1} e_1 (s_l)^{i+1} h_1 (s_r)^{i+1} e_2 (s_l)^{i+1} h_2 \dots (s_r)^{i+1} e_x (s_l)^{i+1} h_x$$

$u_{s_r}, u_{s_l}, u_{h_j}, u_{e_j} \in \{0,1\}^*$

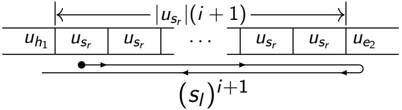


Semi-periodic computations produce regular sets

After time $t(0)$ the sequence S_i is executed between times $t(i)$ and $t(i+1)$ for all $i \in \mathbb{N}$, where $t(i+1) = t(i) + |S_i|$.

$$S_i = (s_r)^{i+1} e_1 (s_l)^{i+1} h_1 (s_r)^{i+1} e_2 (s_l)^{i+1} h_2 \dots (s_r)^{i+1} e_x (s_l)^{i+1} h_x$$

$u_{s_r}, u_{s_l}, u_{h_j}, u_{e_j} \in \{0,1\}^*$



s_l contains m more left moves than right moves

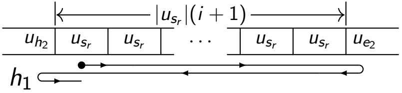
$|u_l| = |u_{s_l}| = m$

Semi-periodic computations produce regular sets

After time $t(0)$ the sequence S_i is executed between times $t(i)$ and $t(i+1)$ for all $i \in \mathbb{N}$, where $t(i+1) = t(i) + |S_i|$.

$$S_i = (s_r)^{i+1} e_1 (s_l)^{i+1} h_1 (s_r)^{i+1} e_2 (s_l)^{i+1} h_2 \ldots (s_r)^{i+1} e_x (s_l)^{i+1} h_x$$

$u_{s_r}, u_{s_l}, u_{h_j}, u_{e_j} \in \{0,1\}^*$

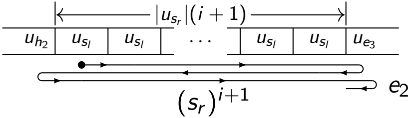


Semi-periodic computations produce regular sets

After time $t(0)$ the sequence S_i is executed between times $t(i)$ and $t(i+1)$ for all $i \in \mathbb{N}$, where $t(i+1) = t(i) + |S_i|$.

$$S_i = (s_r)^{i+1} e_1 (s_l)^{i+1} h_1 (s_r)^{i+1} e_2 (s_l)^{i+1} h_2 \ldots (s_r)^{i+1} e_x (s_l)^{i+1} h_x$$

$u_{s_r}, u_{s_l}, u_{h_j}, u_{e_j} \in \{0,1\}^*$

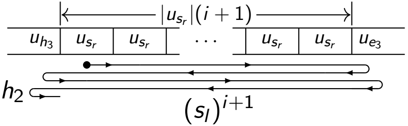


Semi-periodic computations produce regular sets

After time $t(0)$ the sequence S_i is executed between times $t(i)$ and $t(i+1)$ for all $i \in \mathbb{N}$, where $t(i+1) = t(i) + |S_i|$.

$$S_i = (s_r)^{i+1} e_1 (s_l)^{i+1} h_1 (s_r)^{i+1} e_2 (s_l)^{i+1} h_2 \ldots (s_r)^{i+1} e_x (s_l)^{i+1} h_x$$

$u_{s_r}, u_{s_l}, u_{h_j}, u_{e_j} \in \{0,1\}^*$

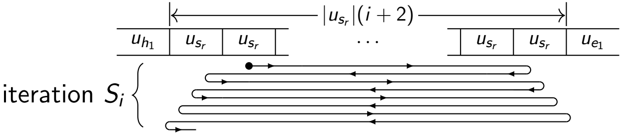


Semi-periodic computations produce regular sets

After time $t(0)$ the sequence S_i is executed between times $t(i)$ and $t(i+1)$ for all $i \in \mathbb{N}$, where $t(i+1) = t(i) + |S_i|$.

$$S_i = (s_r)^{i+1} e_1 (s_l)^{i+1} h_1 (s_r)^{i+1} e_2 (s_l)^{i+1} h_2 \dots (s_r)^{i+1} e_x (s_l)^{i+1} h_x$$

$u_{s_r}, u_{s_l}, u_{h_j}, u_{e_j} \in \{0,1\}^*$

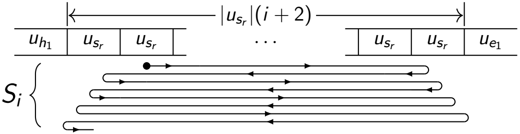


Semi-periodic computations produce regular sets

After time $t(0)$ the sequence S_i is executed between times $t(i)$ and $t(i+1)$ for all $i \in \mathbb{N}$, where $t(i+1) = t(i) + |S_i|$.

$$S_i = (s_r)^{i+1} e_1 (s_l)^{i+1} h_1 (s_r)^{i+1} e_2 (s_l)^{i+1} h_2 \ldots (s_r)^{i+1} e_x (s_l)^{i+1} h_x$$

$u_{s_r}, u_{s_l}, u_{h_j}, u_{e_j} \in \{0,1\}^*$



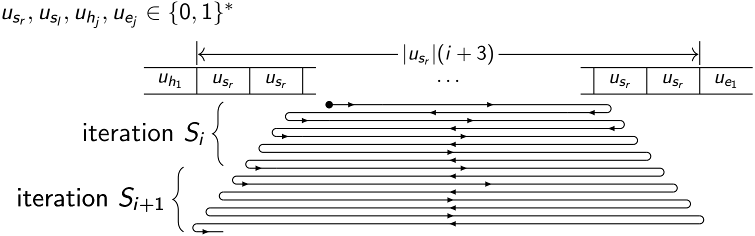
iteration S_i

iteration S_{i+1}

Semi-periodic computations produce regular sets

After time $t(0)$ the sequence S_i is executed between times $t(i)$ and $t(i+1)$ for all $i \in \mathbb{N}$, where $t(i+1) = t(i) + |S_i|$.

$$S_i = (s_r)^{i+1} e_1 (s_l)^{i+1} h_1 (s_r)^{i+1} e_2 (s_l)^{i+1} h_2 \ldots (s_r)^{i+1} e_x (s_l)^{i+1} h_x$$



Pumping the number of u_{s_r} and u_{s_l} words gives all reachable configurations.

Binary counter $M = (\Sigma_5, D_6, Q_3)$ gives regular sets

$$M = \begin{cases} q_a, 0, 0, L, q_a & q_b, 0, 1, L, q_a \\ q_a, 1, 1, R, q_b & q_b, 1, 0, R, q_b \end{cases}$$

M increments a binary number via the 2 steps:

(1) scan right in q_b $1^* \to 0^*$ until $0 \to 1$

(2) scan left in q_a $0^* \to 0^*$ until $1 \to 1$.

Binary counter $M = (\Sigma_5, D_6, Q_3)$ gives regular sets

$$M = \begin{cases} q_a, 0, 0, L, q_a & q_b, 0, 1, L, q_a \\ q_a, 1, 1, R, q_b & q_b, 1, 0, R, q_b \end{cases}$$

M increments a binary number via the 2 steps:

(1) scan right in q_b $1^* \to 0^*$ until $0 \to 1$

(2) scan left in q_a $0^* \to 0^*$ until $1 \to 1$.

Below M increments from 4 to 8.

$$\mathbf{q_a} 1 \underbrace{001}_{4}0 \vdash^2 \mathbf{q_a} 1 \underbrace{101}_{5}0 \vdash^4 \mathbf{q_a} 1 \underbrace{011}_{6}0 \vdash^2 \mathbf{q_a} 1 \underbrace{111}_{7}0 \vdash^8 \mathbf{q_a} 1 \underbrace{0001}_{8}$$

Binary counter $M = (\Sigma_5, D_6, Q_3)$ gives regular sets

$$M = \begin{cases} q_a, 0, 0, L, q_a & q_b, 0, 1, L, q_a \\ q_a, 1, 1, R, q_b & q_b, 1, 0, R, q_b \end{cases}$$

M increments a binary number via the 2 steps:

(1) scan right in q_b $1^* \to 0^*$ until $0 \to 1$

(2) scan left in q_a $0^* \to 0^*$ until $1 \to 1$.

Below M increments from 4 to 8.

$$\mathbf{q_a} 1 \underbrace{0010}_{4} \quad \vdash^2 \quad \mathbf{q_a} 1 \underbrace{1010}_{5} \quad \vdash^4 \quad \mathbf{q_a} 1 \underbrace{0110}_{6} \quad \vdash^2 \quad \mathbf{q_a} 1 \underbrace{1110}_{7} \quad \vdash^8 \quad \mathbf{q_a} 1 \underbrace{0001}_{8}$$

The set of reachable configurations is regular because:

- M generates all possible strings
- left and right scans have a simple form
- left blank word has no effect on the computation
- right blank word effects only a constant number of bits

Conclusion

- For an arbitrary of a 2-state 2-symbol TM with periodic support the set of reachable configurations is regular.
    - First lower bounds given for the minimum possible size of weakly universal TMs (smallest known machines have state-symbol pairs of (4,2), (3,3), and (6,2)).
- The techniques used in this work can be used to prove decidability for other state symbol pairs.
- The table provides and quick and easy way to identify the type of computation of any 2-state 2-symbol TM.
- It would be of interest to provide computational complexity analysis of the TM studied in this work.

Acknowledgements

- Matthew Cook, Institute of Neuroinformatics, University of Zurich and ETH Zurich
- Supported by Swiss National Science Foundation grant numbers 200021-153295 and 200021-166231